The Move to Web Service Ecosystems

Alistair Barros
SAP Research, Brisbane, Australia
alistair.barros@sap.com

Marlon Dumas, Peter Bruza
Queensland University of Technology, Australia
{m.dumas, p.bruza}@qut.edu.au

Web services meet marketplaces

As web services technologies mature, and commercial-scale, service-oriented architectures shift from early adoption to mainstream development, a new revolution of service orientation is emerging. Beyond the orchestration of web services in multi-party business processes, a dedicated treatment for *procuring* web services into different markets is coming into focus.

The first beneficiaries of open procurements of web services are ventures having successfully overcome the dotcom-burst such as Salesforce, StrikeIron, and GrandCentral.[1] These companies leverage XML-based technology to consolidate enterprise application portfolios built by independent software developers for the small to medium markets. Their early successes are paving the way to long-anticipated Amazon/eBay-style marketplaces for web services.

For procurement of services in wider scale, traditional assumptions no longer hold. In larger marketplaces, consumers need a richer exposure of semantics in service descriptions and support of fuzzier search goals. In other words, if services are to be delivered independently, their non-functional properties, such as geospatial and temporal availability, methods of charging and payment, security, trust, rights, and penalties, need to be described in a systematic and precise manner. Furthermore, emerging web service usage scenarios, especially in marketplaces, are pushing the boundaries of coordination and the involvement of intermediaries and reusable service delivery components in end-to-end transactions. Thus, automated assistance will be required by prospective web service suppliers to integrate "instantly" their services with other services, with intermediaries, and with service delivery components. This is to name just a few of the challenges posed by the new frontier of service-orientation.

This paper presents a vision for the next generation of service-oriented systems, which we term *web service ecosystems*. The paper reviews early manifestations of these systems and identifies challenges that will need to be addressed on the road to the web service ecosystems vision.

The rise of web service ecosystems

Service-oriented architecture (SOA) has gained mainstream acceptance as a strategy for consolidating and repurposing legacy applications to be combined with new applications in more dynamic environments, through self-contained, reusable, and configurable services. As fostered through the web services standards stack, servicewies, once in place, can

[1] http://www.salesforce.com, http://www.strikeiron.com, and http://www.grandcentral.com.

interoperate with other services, be composed into long-running business processes, spanning intra- and inter-organizational boundaries, and be procured through different business domains and market sectors.

As web services are exposed and connected with one another, they give rise to *service ecosystems*. In these ecosystems, services are deployed, published, discovered, delivered to different business channels through specialist intermediaries (e.g., payment, authentication, and mediation services), and monitored. A groundswell of web service ecosystems is emerging on the "dotcom" front, and targeting enterprise applications for the small to medium sector. Following the successes of Amazon and eBay goods marketplaces, the first significant efforts of bringing together web services using more open procurement models are precipitating the long-anticipated internet service marketplace concept.

Salesforce.com, with around a half million subscribers, recently announced *AppExchange*[2] to expand its CRM software base into a full business software portfolio. To achieve this, testing and deployment tools are available through AppExchange for developers who code against Salesforce.com services. Around 50 services are already available, and expectations are growing that AppExchange will foment a comprehensive enterprise applications marketplace.

The benefits for Salesforce are obvious. It is set – at least in principle – to absorb the "fittest" of compliant software components from external developers into its newer releases. At the same time, Salesforce is encouraging further avenues of procurement through open publication of its web services and configuration for integration with other components. This reduces some of the complexity of procuring these services through new applications. *NetSuite*, another of a growing number of on-procurement initiatives, integrates Salesforce services with its ERP and portals solutions, thereby competing with Salesforce's value-added layer.

More sophisticated procurements can be seen through more general service marketplace solutions, resembling public sector online services initiatives like *UKOnline*.[3] *StrikeIron's* repertoire of service delivery functions like customer account management, wizard-based software-as-a-service publishing, search/discovery, purchasing, authentication, billing, payment, and systems monitoring is supplemented through third parties and established intermediaries, e.g., PayPal.[4] A service provider is able to service-enable, deploy, and deliver endpoints to different business channels without having to factor in hosting and service delivery functionality. Indeed, Salesforce services like tele-call listing are available through StrikeIron, and StrikeIron's partnership with Dreambuilder enables value-added extensions through U.S. government services for address, phone number, and taxation type of verifications.

In addition to marketplace support, new ventures are also vying for growth by addressing quality-of-service interoperability, typically available in larger commercial products. *GrandCentral's Business Service Network* underpins message exchanges between hosted business application suites, like ServiceObjects, RecruitForce.com, and SPS Commerce, for EDI hosting with reliable messaging, identity management, and other security services. Thus, GrandCentral supports more sensitive transactions, like Xignite's financial services with

---

[2] "An eBay for Business Software", Business Week Online, 19 September 2005: http://www.businessweek.com/magazine/content/05_38/b3951097.htm

[3] Now superseded by DirectGov: http://www.direct.gov.uk

[4] http://www.paypal.com

StrikeIron (and therefore Salesforce) services, moving service composition beyond basic-lookups into message interactions in the style of Rosetta Partner Interface Protocols (PIPs).[5]

The impact already launched by Salesforce, StrikeIron, Grand Central, and others, can be summed up simply – the web. Much as the web revolutionized the worlds of information and goods, so the first web service ecosystems are providing an early indication of internet commerce for services. Greater sophistication of web service ecosystems can be expected as the internet giants make their move. In July 2005, Amazon filed a patent application for “web service marketplaces.” eBay hosts a growing number of vertical service using ChannelAdvisor[6] portals (and is extending auctioning and subscriptions features of its goods marketplace to services). Google plans to harmonize the web by unifying information, desktop/office, geospatial, social networking, and service worlds.

Taken together, these developments herald the first wave of mass growth in web services, which, according to estimates from market intelligence agency IDC, will be $9.1 billion by 2008. This correlates to significant upward trends in service outsourcing[7] and national strategic investments for longer-term development of web services economy[8].

Obstacles to web service ecosystems

As seen through the previous section, *web service ecosystems can generally be described as a logical collection of web services whose exposure and access are subject to constraints, which are characteristic of business service delivery.* A lower level analogy can be found with application servers, where access and inter-operation of components is regulated by middleware functions such as discovery/brokering, remote access, object pooling, transactions, asynchronous messaging, persistence management, and so on.

For web service ecosystems, the regulation entails service delivery at the *business* level. Therefore, frameworks supporting web service ecosystems constrain the way services are discovered, authenticated, adapted and mediated, transacted, charged and paid for, monitored, penalized against breaches of use, fulfilled, and ranked. This is from the service demand side. From the supply side, regulations would be imposed on the way services are published, value-added or repurposed through composition with other services, re-provisioned through leasing and licensing, and independently brokered. The other fundamental difference with application servers is that the “middleware” comprises web services themselves that are

---

[5] http://www.rosettanet.org

[6] http://www.channeladvisor.com

[7] According to Gartner, Accenture and Aberdeen Group, 30-80% of corporate expenditure in F1000 companies (roughly $1 billion per company) lies in service outsourcing obtained through online listings, with commercial printing, marketing research, and contract labour among the largest categories. In the U.S. Healthcare industry, it is estimated that over 70% of new software licences will be sold as software-as-a-service.

[8] Two government strategies are noteworthy. Firstly, following the success of UKOnline, the British Government is investigating a pilot with private sector involvement of a large-scale marketplace for community services like domestic rentals, childcare, and temporary labour. Its “National E-Markets” study cites the steep growth in online purchases, for example: nearly 3 million Britons bought something online in 2004; UK eBay sales were $100 million in 2004, up 52% from the previous year.

Secondly, the U.S. Office of Management and Budget (OMB) embarked on one of the largest marketing campaigns for promoting pervasive e-government services over 2004/5. Its focus is on the Bush administration's 25 "Quicksilver" e-government initiatives, which include targeting particular customer segments, innovative ideas on how to increase usage, and methods on providing greater synergy among e-government offerings.

potentially outsourced in the ecosystem. Thus, a variety of payment "engines" with specialization in particular payment methods could be registered so that they can be bound to instances of running services requiring particular payment functionality.

A service marketplace can then be seen as a specific, and currently prominent, example of a web service ecosystem. A "pan-marketplace," spanning multiple marketplaces, as in multi-jurisdictional marketplace of government online services, is an extrapolation of this[9]. General-purpose supply/value chaining, aka business service networks, is yet another example.

In summary, web service ecosystems trans-locate not so much web services, but web service delivery – collectively making more explicit the notion of service procurement. Clearly, the extent to which services delivery is regulated is a moving target, and will develop as Internet business models[10] grow in sophistication.

In the following sections, a number of critical aspects of service delivery are discussed to gauge current limitations of web service ecosystems. These are

- **Service supply and distribution networks:** the extent to which different service supply and distribution roles are supported;
- **Service discovery and planning:** the flexibility of capturing semantics of services in wider, domain non-specific ecosystems, and the flexibility of service discovery in this setting, including support of fuzzy search goals;
- **Conversational service interactions:** the extent to which service interactions go beyond basic request-to-execute transactions, in support of negotiations and long-running, multi-party service interactions;
- **Service quality management:** the extent to which non-functional properties are captured in service descriptions such that the quality of delivery can be managed in accordance to service level agreements;
- **Service mediation and adaptation:** the extent to which automated assistance is available to repurpose services in order to compose them and bind them to service delivery components in ways not foreseen when these services were originally designed.

Below, we discuss each of these aspects in turn.

Service supply and distribution networks

The underlying business models of current web service ecosystems are mostly about bringing service consumers closer to service providers, or allowing services to be accessed through intermediaries with collected revenue then passed back to providers. WSDL enablement through software-as-a-service wizards is the mechanism for allowing services to be published in different service deployments. Current forms of service supply and distribution networks are rather limited, compared to commercial practice, as well as to the newer models that inevitably lie ahead.

[9] In Australia, such a vision was raised through the Federal government's Bizdex (www.bizdex.com.au) initiative.

[10] For a discussion on business models for web service delivery, see http://digitalenterprise.org/models/.

Figure 1 illustrates the inter-play between different service delivery roles in a decoupled service delivery setting.

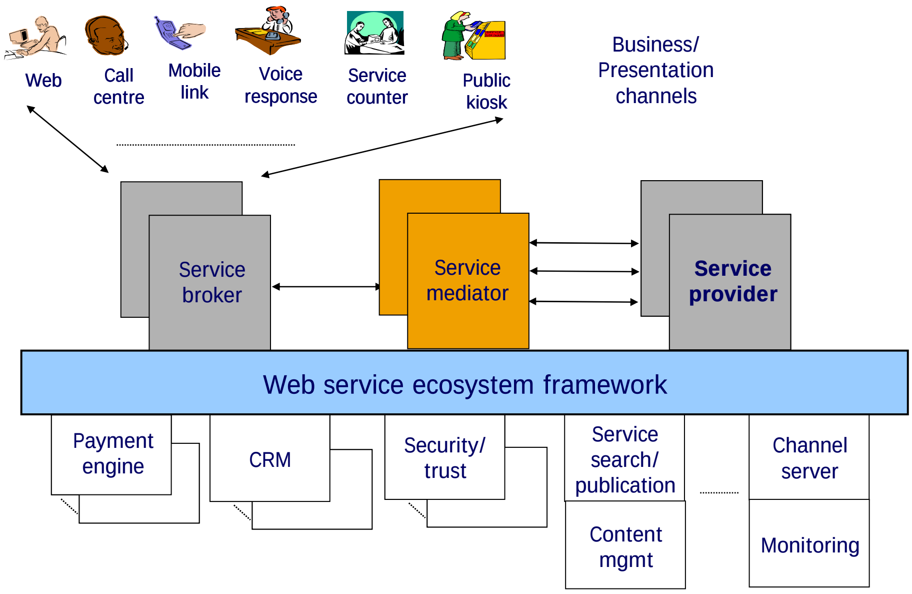

Figure 1. Service supply and distribution supported in a web service ecosystem framework

The most obvious point of decoupling is that between service providers and service brokers. In order to increase the procurement of services through different markets, service providers outsource the "front-desk" role to third party brokers (which may operate outside jurisdictional and national boundaries of service providers). The "backend" responsibility of the service still remains with the service provider, while a broker delivers the service through different business channels (e.g., shopping centers, web portals, GPS applications). As such, a service broker is responsible for delivering the service in accordance with constraints imposed by the service including authentication; payment; timeliness and quality in collecting input and passing back output; flexibility of service presentation conducive to business channels, without compromising essential service functionality; monitoring service consumption and enforcing penalties; etc. A brokered service will also impose further constraints of delivery, such as availability and incentive mechanisms for using services (e.g., frequency flyer points).

The separation of service provider and broker brings up the issue of routing between the two, and, more generally, routing that involves several service providers. Considering the heterogeneity of services, interactions need to cross well-adapted service interfaces. The greater the number of services involved and the more complex the adaptation, the more the challenge shifts to service *mediation*.

Yet a further extension of service supply and distribution is the role of service provider. Through software-as-a-service support, new providers could come on board through service leasing, licensing, and other forms of custodianship that are subject to terms of agreement. In these instances, the functionality is "copied," and a new endpoint is created.

Consider now the support for the kind of supply and distribution roles discussed above. Service brokerage is fixed, with intermediaries like StrikeIron also acting in the role of service broker for service providers. In effect, only one broker is available to deliver the services, leaving only that entity to control which business channels should serve up the services. Variation of business channels occurs when the services permeate through different ecosystems (as is the case for StrikeIron indirection of Salesforce); however, the conflation of the points of service deployment and brokerage is apparent. In turn, the role of service mediation, is also “wired” into entities controlling the ecosystem.

Seamless ways of opening up the service supply and distribution networks – from service provider to service broker to potentially open-ended third parties with their specific delivery constraints (e.g., methods of payment supported, availability of service delivery) – are necessary to embrace efficiently the emerging models of internet commerce. In the same way that the proliferation of web content and web applications prompted the adoption of novel business models such as pay-per-click and, more generally, infomediaries, new business models based on the notions of referral, delegation, and intermediation, are poised to emerge in the context of service ecosystems. Accordingly, the current web service infrastructure will need to evolve in order to incorporate mechanisms to support these business models. In particular, request referral and relaying will have to be promoted to first-class citizens as service providers will focus on their core competence and refer or relay ancillary requests to third parties.

While Salesforce, StrikeIron, and the like are burdened with building up a critical mass of ERP functionality and robustness of interoperability, a window of opportunity exists to focus efforts on more systematic support for web service ecosystems. Existing ERP infrastructures such as SAP’s Enterprise Services Architecture (ESA) already have a comprehensive set of decoupled enterprise applications that could be marshaled into web service ecosystems not only as service endpoints but also to support underlying service delivery roles. However, a dedicated tier needs to be built on top of these systems through which third parties on a service supply and distribution chain would become instant brokers, mediators, specialist intermediaries (e.g., payment engine, CRM), leased/licensed custodians, or hold other emerging service delivery roles. Virtually no formalization exists for service leasing or similar licensing models.

Service discovery and planning

Current provisions for discovery are based on keyword searches through repositories. Keywords are nominated by service providers through publication and advertising features of software-as-a-service functions. Details of message inputs, outputs, and methods are also captured from WSDL file scans and factored into searches.

Such discovery techniques are suitable in tightly coupled and well-scoped domains where service consumers can determine what services offer and how they can be independently utilized from search results. In other words, users are expected to know what they want before they search.

However, within the setting of more widespread web service ecosystems involving greater heterogeneity, this assumption breaks down. The wider the domain of an ecosystem – like an Amazon marketplace – the more general search schemas are. Therefore, the greater onus is on suppliers of services to enrich service descriptions that can be queried for the different variety of contexts in which services can be used. The current schemes of service classification on offer are noticeably simple by real-world standards. Consider for instance that only basic details of non-functional properties of services are available through current schemes[11].

Even if sophisticated schemes were available, as envisioned in ontology-based service semantics of Semantic Web technologies, it is doubtful whether overarching committees can reliably arbitrate service descriptions with sufficient foresight of the possibilities in which services would be utilized. Committees have limited insights and their utility, paradoxically, lies in securing common-denominator consensus for ontological terms and references.

Left by the wayside are large sources of textual documentation about services, in regard to their business strategic, tactical, legal, or legislative, communal, jurisdictional, and demographic contexts – just to name a few possible types of service-related documentation. Sources of service knowledge dispersed through the environment in which services operate – jurisdictions, business missions, consumption points, and so on – can open up the variety of known and unknown contexts of services (a term known as *latency semantics* in the field of cognitive science).

In service ecosystems, this knowledge could be used outside the traditional bases of services, to determine how services could be procured through the service supply and distribution networks. Service providers could determine how to repurpose their applications in the variety of marketplaces. For instance, more competitive packaging of application components through service endpoints might be determined. Similarly, service brokers could determine through which marketplace channels to target services, how demand could be driven up, what incentives to adopt, and so on.

To enable web service ecosystems, more suitable free-text search techniques would provide a strong competitive advantage. Whereas current ecosystems are expected to use UDDI-style repositories with keywords-based search, there is great potential in providing a suitable combination of free-text search techniques with ontology-based search techniques. This would furnish web service ecosystems with structured *discovery* as well as unstructured information retrieval style searches typical in service *planning*. Service planning – an earlier and more iterative phase of conventional service discovery – implies uncertainty of goal in conducting searches and a search agenda that becomes clarified with each search iteration.

Increased knowledge from a variety of sources captured and indexed through automated techniques opens up the possibility for search *agents* to guide users from a fuzzy starting point (e.g., "How do I open up a coffee shop/") through the variety of contexts (regulated registration tasks, market analysis, logistics of resourcing a business, etc) to then hone in on the users' agenda. Endowed with a Google-like query browser, this would be a powerful tool for service consumers' service supply and distribution roles – and, conceivably, even in the formation of marketplaces.

Conversational service interactions

Internet commerce has created newer forms of service interactions than traditional marketplace transactions. Amazon/UKOnline, single-consumer-to-service transactions – e.g., making customer listings, doing basic look-ups and verification checks, and purchasing goods – are giving way to more distributed, pull-oriented and data streaming modes of interaction on the web. Marketplace auctions, voting, and subscription-based RSS feeds are

[11] A detailed classification of business services, featuring non-functional service properties, can be found in http://www.service-description.com.

enhancing wider spans of participants and semi-structured, audio and video data in conventional transactional forms.

In larger environments with long-running transactions in supply/production chains, process-based web services *orchestration* techniques are utilized. WS-BPEL[12] is a developing standard with support in various commercial tools such as Oracle BPEL, Microsoft BizTalk, and SAP's ccBPM tool.[13] It allows the ordering of web service interactions to be captured through single-party process models. External interactions are captured through outgoing and incoming messaging operations in the model. In addition, support for multi-party collaborations through global model definitions has received attention in the context of W3C's Web Services Choreography Description Language (WS-CDL) initiative.[14]

Despite the fact that web service ecosystems are garnering large-scale and diverse exposure of services, executable modeling languages like WS-BPEL, ebXML/BPSS, WSCL, WSCI, and the more recently developing WS-CDL, have not as yet been exploited in these environments. Greater flexibility and productivity are available for service providers to enrich services in collaboration with other services providers through *graphical* service compositions. Even more radical, graphical tools could foster personalized, end-user composition of web services.

As the value of process composition filters out to the "dotcom" world, certain challenges will have to be overcome in order to harness orchestrated and choreographed service interactions with marketplace-style transactions. Does the single point of payment still hold for long-running processes? Will one-off authentication still hold? It is doubtful, considering that different paths with different service provisions can be taken through branching in processes. Different payment points – indeed, pay-as-you-go – should be expected. Such transactions could not be expected to cover single sessions, and more systematic ways of handling authentication and access control need to be factored. From a security perspective, longevity of process leads to the issues of trust chains and delegated access. How critical security considerations can sit seamlessly with BPM is still the subject of ongoing research.

The issue of assimilating process composition into "dotcom" transactions provides a touchstone in further considerations symbiotic with the web. The first generation of service provision on the web has seen a stove piping between service interactions and information search. Information relevant to services, as they are being utilized, is dispersed on the web. The bridge yet to be crossed for service ecosystems is how information ecosystems can be tapped to filter out timely and relevant information associated with a running service.

In general, increased efforts to harmonize different realms on the web can be expected. Google has signaled its pursuit of bringing greater coherence of information, service, geospatial and temporal sensitivity, and desktop/office. At the same time, initiatives like Digital City are expanding the service and information ecosystem reach to material interaction. Motivating applications include vehicle guidance, remote access to public instruments, and home devices (electricity meter readings) from utility service providers.

---

[12] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel

[13] For a list of implementations of BPEL, see: http://en.wikipedia.org/wiki/BPEL. For information about the products cited above, see, e.g.: http://www.oracle.com/technology/products/ias/bpel, http://www.biztalk.org, and http://help.sap.com/saphelp_sem40bw/helpdata/en/3c/831620a4f1044dba38b370f77835cc/content.htm.

[14] http://www.w3.org/2002/ws/chor

Service quality management

In wider spanning service ecosystems, several service providers may offer functionally replaceable services that differ in their extra-functional characteristics, such as usage terms and quality of service delivery. Service providers need to be responsive – potentially in real-time – to negotiate variations of service delivery requirements (e.g., price, deliverable timetable). Service ecosystems should therefore explicitly support the negotiation process, reducing non-critical human involvement and providing decision-makers with the information they require to formulate and assess service offers.

In particular, automated support for negotiation over services is needed for comparing requirements and preferences of prospective service users against capabilities and terms of usage of service providers. This calls for languages and tools supporting the capture of non-functional, business-oriented service properties, including: temporal and spatial availability, pricing models, payment mechanisms, trust, reputation, promises, penalties, escalation, and dispute resolution mechanisms, to name just a few.

In many industry sectors, service contracts are in place that include precise definitions of service reliability and responsiveness guarantees, and penalties that apply when these guarantees are breached. Capturing these contracts in a machine-understandable way allows the associated guarantees and penalties to be monitored and enforced automatically, and facilitates the comparison and matchmaking of service offerings with respect to customer requirements.

Techniques for matching customer requirements and preferences, against possibly parameterized service offers, can build upon explicit representations of such properties. Making these non-functional service properties explicit is also *a sine qua non* condition to formally capturing service level agreements and, more broadly, service contracts. Being able to link these agreements to collaborative process coordination models allows these contracts to be automatically monitored.

Service mediation and adaptation

The assimilation of services through service ecosystems presents major integration development and maintenance costs. Service providers need to compose their services effectively in coordination with other services if they are to engage in oncoming market opportunities and situations. Further up the supply and distribution chain, if services are to be brokered and delivered through other intermediaries (e.g., for authentication, payment, device-specific service presentations), they will need to be interfaced with service delivery components that operate in various ways. Thus, one can expect that services will have to interact with one another in ways not necessarily foreseen during their development or deployment. A key challenge in this setting is service mediation: the act of repurposing existing services so that they can interact in unforeseen manners by intercepting, storing, transforming, and routing messages going into and out of these services.

A prominent sub-problem of service mediation is that of service interface adaptation [3], where the goal is to keep interfaces as generic as possible while adapting to functions peculiar to implementation or prone to change. As a basic example, consider a procurement service which, after sending a Purchase Order (PO) to a supplier's order management service, expects to receive one and only one response. Now, consider the case where this procurement service is required to engage in a collaboration wherein the order management service may send a first response acknowledging the PO and accepting or rejecting a subset of its line items, possibly followed by one or more additional updates to accept or reject the remaining line items as their availability is determined. The mismatch between the “provided” (i.e., “as is”) and the “required” (i.e., “to be”) service interfaces is illustrated in Figure 2.

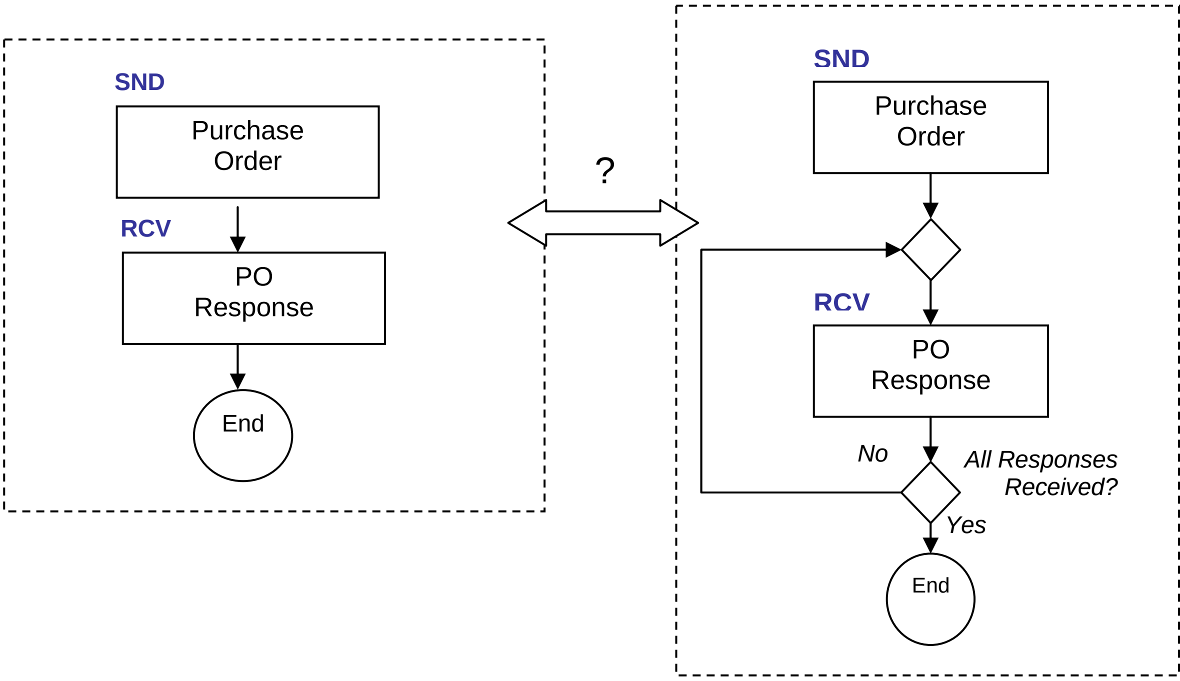

Figure 2. Mismatch between provided and required service interfaces

More generally, a service may be required to participate in multiple collaborations such that in each of them, a different interface is expected from it. Implementing, testing, deploying, and maintaining adapters to deal with the multiplicity of interfaces required from a service may be costly and error-prone. This calls for specialized tool support based on high-level concepts, as opposed to implementing adapters using programming languages extended with basic message manipulation primitives.

Service interfaces cover structural aspects, such as message types and supported transport protocols, but also behavioral aspects, such as ordering dependencies between messages as captured in, e.g., BPEL business protocols. The problem of service interface adaptation from the structural perspective has received considerable attention, leading to design-level mapping tools such as Microsoft BizTalk Schema Mapper or SAP Integration Builder's Mapping Editor. In comparison, the problem of service interface adaptation from a behavioral perspective has received less attention. SAP’s ccBPM tool supports behavioral interface adaptation through multi-mappings and message bundling patterns, but these mechanisms only address a restricted set of behavioral interface adaptation scenarios.

Outlook

The convergence of internet marketplaces with service-oriented architectures is opening an era where software functionality, on the one hand, and software adaptation, delivery, and distribution, on the other, will be increasingly treated as orthogonal aspects on par with each other. Enterprise software functionality is likely to be commoditized, thus forcing vendors in this space to expand their user bases through larger-scale reuse. On the other hand, software adaptation, delivery, and distribution, as well the human-based support processes that underlie these activities, will provide new revenue streams and reward innovative practices. In this setting, the potential value of software will be, to a large extent, determined by its incorporation into service offerings, especially over the web, and the ability for these services to participate in existing and new service ecosystems. In these environments, the norm could be for services to be used and bundled in ways not originally foreseen, and how well a service is inter-connected with others will be a measure of its success, in a way similar to how, in the "traditional" web, the ranking of a web page is determined in great part by its connections to other pages.

However, there is a bumpy road ahead towards the web service ecosystems era. The explicit and formal description of non-functional aspects of services, such as guarantees, pricing, payment, penalties, and delivery modes, may become a bottleneck unless a better understanding, possibly leading to standards, is developed. Also, while current web service technology and standards focus on simple bilateral request-response interactions, web service ecosystems will push the limits by requiring long-running, conversational, and possibly multilateral interactions.[15] Finally, mediation needs to be lifted to a first-class role. But, unlike traditional mediators, mediators in web service ecosystems will not be limited to dealing with data heterogeneity and formatting issues. Instead, they will need to deal with mismatches between long-running interactions wherein events are inter-related in complex ways. Also, mediators may need to deal with non-functional aspects, for example, by detecting potential service guarantee violations and triggering failovers or other escalation mechanisms. In service ecosystems, developing and deploying mediators and adapters might become as important as linking content in the first-generation web.

[15] For a discussion on complex interaction patterns, see http://www.serviceinteraction.com